

REMARKS

Claims 1-33 are pending in the present application. Claims 1-5, 23-27 and 31 have been amended, and Claims 32 and 33 have been added, herewith. Reconsideration of the claims is respectfully requested.

Amendments were made to the specification to correct errors and to clarify the specification. No new matter has been added by any of the amendments to the specification.

I. 35 U.S.C. § 102

The Examiner rejected Claims 1, 3, 5-6, 23-26 and 31 under 35 U.S.C. § 102(e) as being anticipated by Stracovsky, US Pat. No. 6,587,894. This rejection is respectfully traversed.

With respect to Claim 1, Applicants show that the cited reference does not teach the claimed steps of "determining if the command node in the first queue collides with the command node in the second queue", and "if the command node in the first queue does not collide with the command node in the second queue, then moving the command node in the first queue from the first queue into the second queue". In rejecting Claim 1, the Examiner cites Stracovsky's abstract and Col. 3, lines 10-44 as teaching these claimed steps. Applicants show that Stracovsky states in the abstract:

"According to the present invention, a system for reordering commands to achieve an optimal command sequence based on a target response restriction is disclosed. A data queue coupled to a command queue is arranged to store a time indicating when the data transfer will appear on the data bus between the controller for an already issued request to the target device as well as arranged to store the burst bit and the read/write bit (r/w). The system also includes a collision detector coupled to the data queue and the command queue arranged to detect the possible collisions on the data bus between the issued command that is stored in the command queue and already issued commands that are stored in the data queue. A queues and link controller is coupled to the collision detector and the data queue and the command queue and is arranged to store and reorder commands to be issued wherein the controller calculates the new issue time of commands as well as a time when the data appears on the data bus."

The details of this store and reorder of commands is described at Stracovsky Col. 25, line 60 – Col. 26, line 38 and Figure 20. There, Stracovsky states:

"FIG. 20 illustrates a collision detection system 2000 that is another implementation of the collision detection system 1500 shown in FIG. 15. In this embodiment, the collision detection system 2000 reorders commands to achieve an

optimal command sequence based on target response restrictions and determines the optimal slot for data transfer between initiator controller and target subsystem. Because the reordering of the commands can not cause collision of the different data packets on the data bus, a collision detector 2002 that prohibits to the issuance of a particular command if the command data transfer related to this particular command would cause data conflict is required. In the describe embodiment, the collision detection system 2000 includes the collision detector 2002 that is coupled to a command queue 2004.

In the described embodiment, the collision detector 2002 detects all possible data collisions between a "to be issued" command (that is stored in a command queue 2004) and "already issued" commands (that are stored in a data queue 2006). In the described embodiment, there are N command queues 2004 each being coupled to a multiplexer 2008. Each of the N command queues 2004 are arranged to store those commands that are to be issued, a time factor "d_time_{ND}", indicating when the data transfer will appear on a data bus between the universal controller 104 and the target device (i.e., shared resource) 108 after the command was issued to the target device, a burst-bit (b_{ND}) indicating data burst transfer, and a read/write bit (rw_{ND}). In the described embodiment, the data queue 2006 stores a time factor "d_time_D" indicating when the data transfer will appear on the data bus between controller 104 and the target device 108 for an already issued request to the target device. The command queue 2006 also stores the burst-bit (b_{ND}) and the read/write bit (rw_{ND}).

In a preferred embodiment, the collision detection system 2000 includes a queues and link controller unit 2010 arranged to store and reorder those commands that are to be issued. The queues and controller unit 2010 also calculates the new issue time of commands and a time when the data appears on the data bus. The queues and controller unit 2010 also transfers the issued element from the command queue into the data queue as well as removing it from the command queue after the command was issued. The queues and controller unit 2010 also removes data elements from the data queue after the access to the memory has been completed."

As can be seen, there are a plurality of command queues 2004 containing commands to be issued. Each of these command queues is coupled to a multiplexer 2008, which is controlled by queue and link controller 2010 (indicated in Figure 20 as element 2008, an apparent typographical error). Upon detection of a collision between commands by the queue controller, the order of commands to be executed are reordered by the queue and link controller by selectively passing a command from a particular command queue 2004 using multiplexer 2008 which outputs the appropriate command at its output. Of particular noteworthiness is the fact that this reordering of commands is accomplished by selecting one of a

plurality of commands from each of a plurality of command queues 2004 using multiplexer 2008 as controlled by queue and link controller 2010.

In contrast, Claim 1 recites "determining if the command node in the first queue collides with the command node in the second queue", and "if the command node in the first queue does not collide with the command node in the second queue, then moving the command node in the first queue from the first queue into the second queue" (emphasis added by Applicants). The cited reference does not move commands between queues based upon a determination of whether a command node in a first queue collides with a command node in a second queue, as claimed. Thus, Claim 1 is shown to not be anticipated by the cited reference.

Applicants initially traverse the rejection of Claim 3 for similar reasons to those given above regarding Claim 1, of which Claim 3 ultimately depends upon. Applicants further show that the cited reference does not teach or otherwise suggest any use of a Rotational Positioning Sorting (RPS) algorithm as a routine used to sort a queue, as claimed. Nor has the Examiner alleged any such teaching. Thus, Claim 3 is further shown to not be anticipated by the cited reference.

Applicants traverse the rejection of Claims 5-6 for reasons given above regarding Claim 1, of which Claims 5 and 6 depend upon.

With respect to Claim 23, such claim recites three queues: a first queue comprising a command node, a second queue comprising a command node selected from the first queue, and a third queue comprising a command node selected from the second queue according to a predefined optimization scheme. While the cited reference teaches a plurality of commands queues (see, e.g., Stracovsky Figure 20, elements 2004), none of these queues comprise command nodes selected from others of the queues, as claimed. Rather, new commands are input to each of the respective queues from a common input labeled "new command", and the output of each command queue is coupled to multiplexer 2008. There is no ability for inter-queue selection of commands, where a second queue comprises a command node selected from a first queue, and a third queue comprises a command node selected from the second queue. Thus, Claim 23 is not anticipated by the cited reference.

Applicants traverse the rejection of Claims 24-26 for similar reasons to those given above regarding Claim 23, of which Claims 24-26 ultimately depend upon.

With respect to Claim 31, such claim has been amended to recite that the queue processing means controls the position and flow of command nodes within and between the plurality of command queues. As described above with respect to Claim 1, and shown specifically in Stacovsky's Figure 20, the cited reference has no ability to control flow of command nodes between command queues. Thus, amended Claim 31 is not anticipated by the cited reference.

Therefore, the rejection of Claims 1, 3, 5-6, 23-26 and 31 under 35 U.S.C. § 102(e) has been overcome.

II. 35 U.S.C. § 103, Obviousness

The Examiner rejected Claims 27-30 under 35 U.S.C. § 103(a) as being unpatentable over Stracovsky, US Pat. No. 6,587,894. This rejection is respectfully traversed for similar reasons to those given above regarding Claim 23.

Therefore, the rejection of Claims 27-30 under 35 U.S.C. § 103(a) has been overcome.

III. Allowable Subject Matter

A. The Examiner objected to Claims 2 and 4 as being dependent upon a rejected base case, but stated that such claims would be allowable if rewritten in independent form including all limitations of the base claim and any intervening claims. Applicants have amended such claims accordingly.

B. Applicants graciously acknowledge the allowance of Claims 7-22.

IV. Newly Added Claims 32 and 33

Claim 32 and 33 have been added herewith, and examination of such claims is respectfully requested.

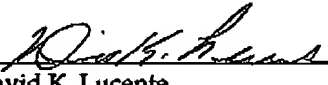
V. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

SEAGATE TECHNOLOGY LLC
(Assignee of Entire Interest)

Date

1/7/04


David K. Lucente
Seagate Technology LLC
Intellectual Property Dept. - COL2LGL
389 Disc Drive
Longmont, Colorado 80503
(720) 684-2265 (telephone)
(720) 684-2588 (facsimile)